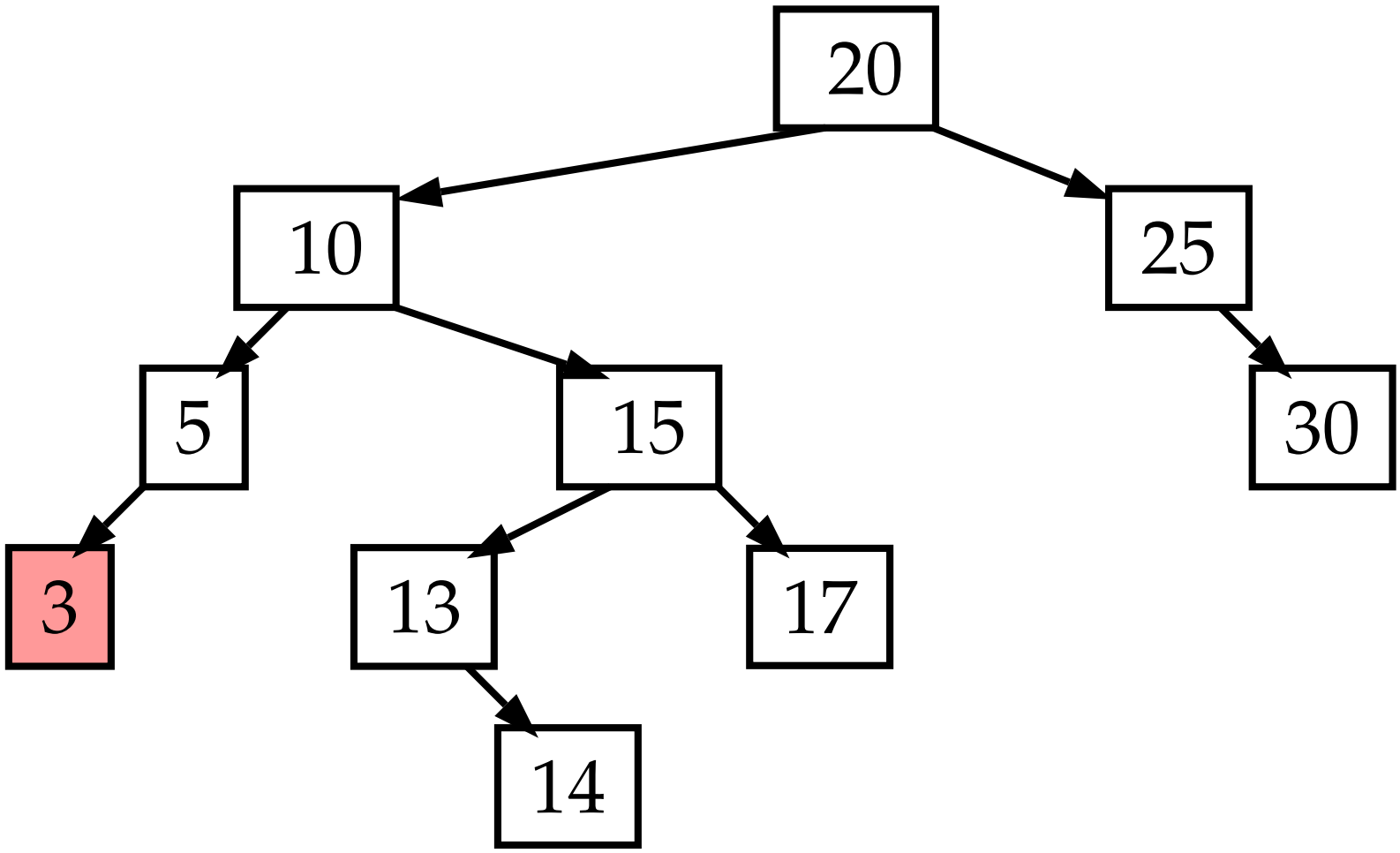
INF1010 - Puslegruppa

Selvbalanserte binærtrær!

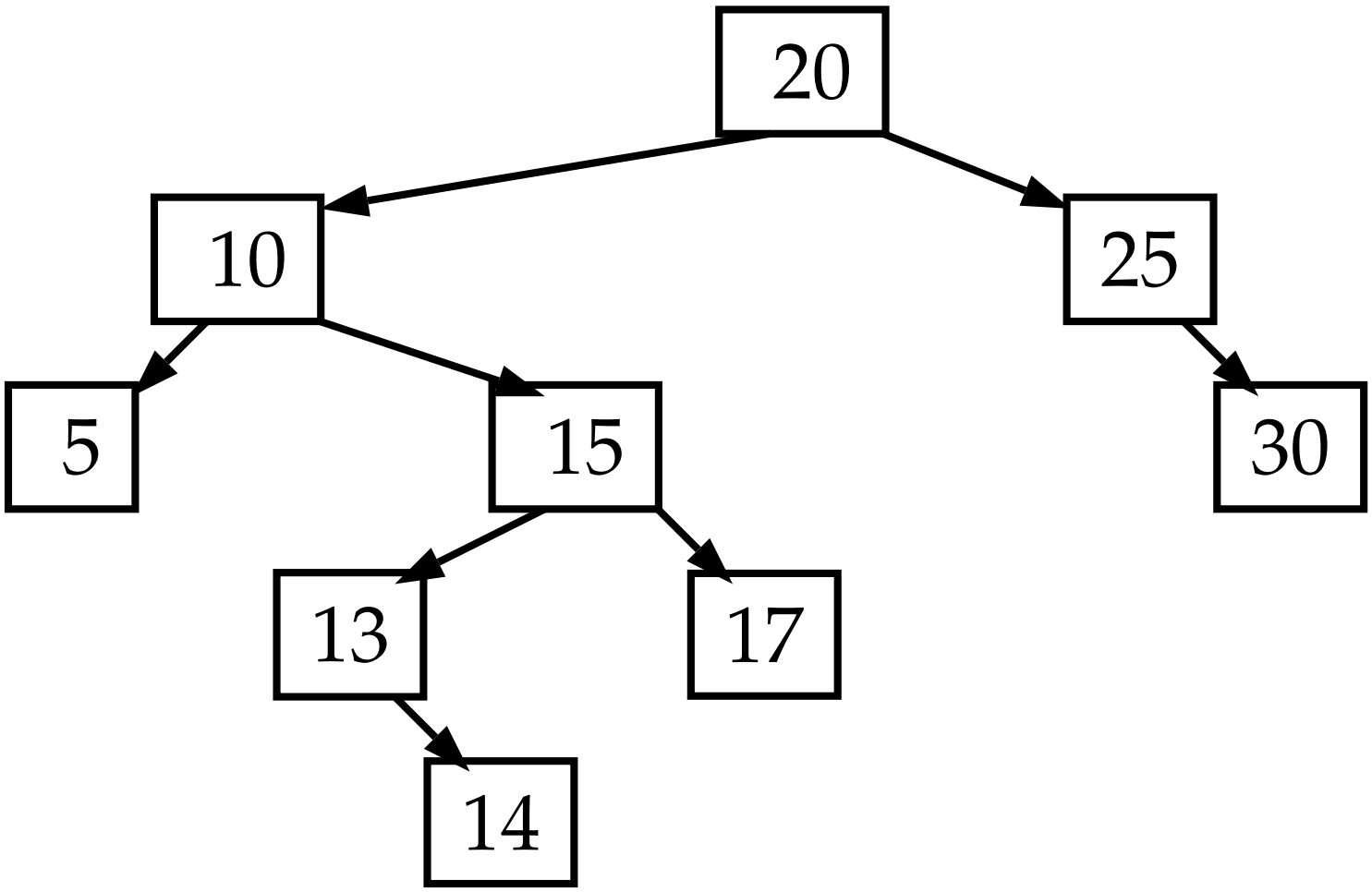
Simen Heggestøyl – simenheg@ifi.uio.no
Sigmund Hansen – sigmunha@student.uio.no

17. mars 2011

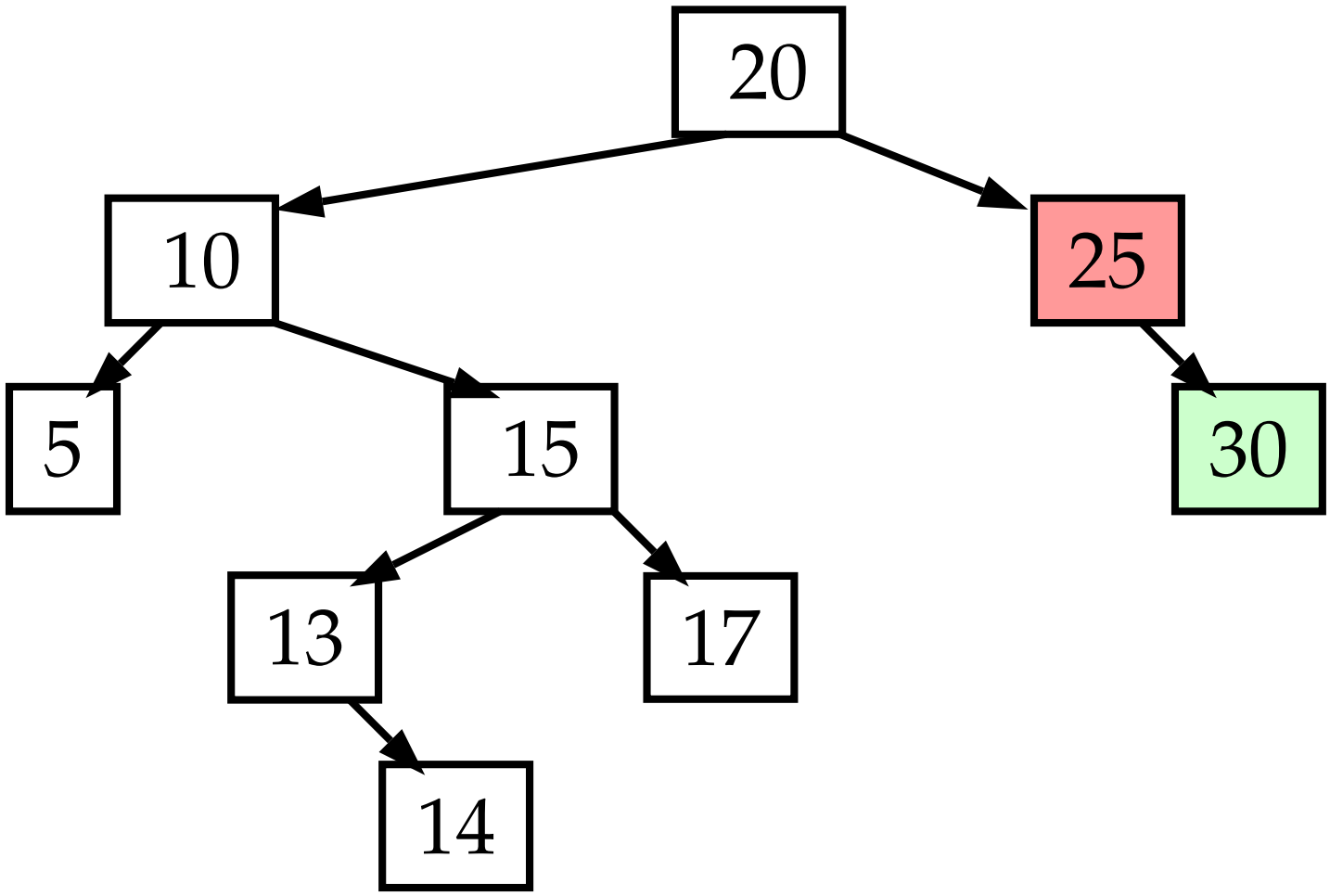
Sletting - løvnode



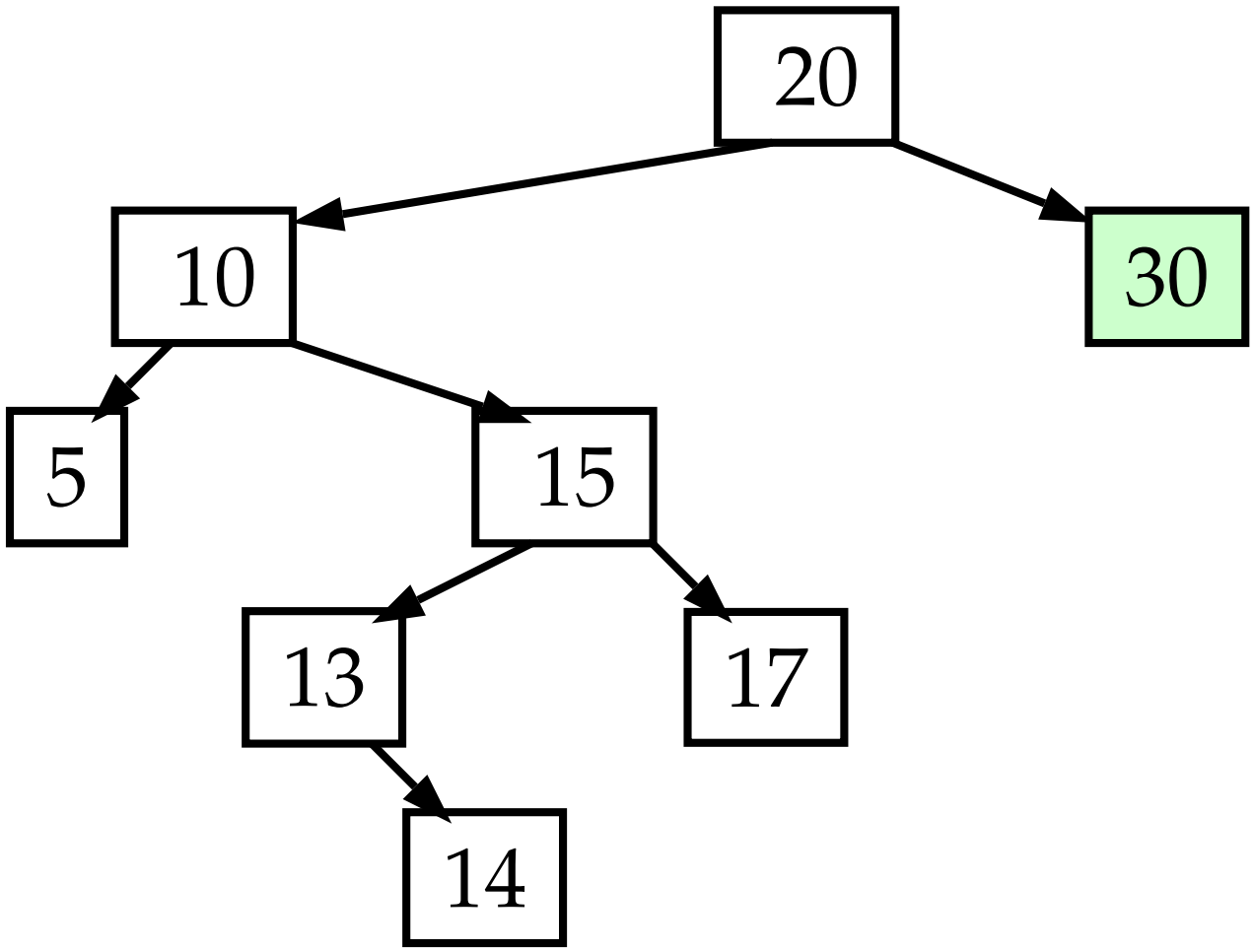
Sletting - løvnode



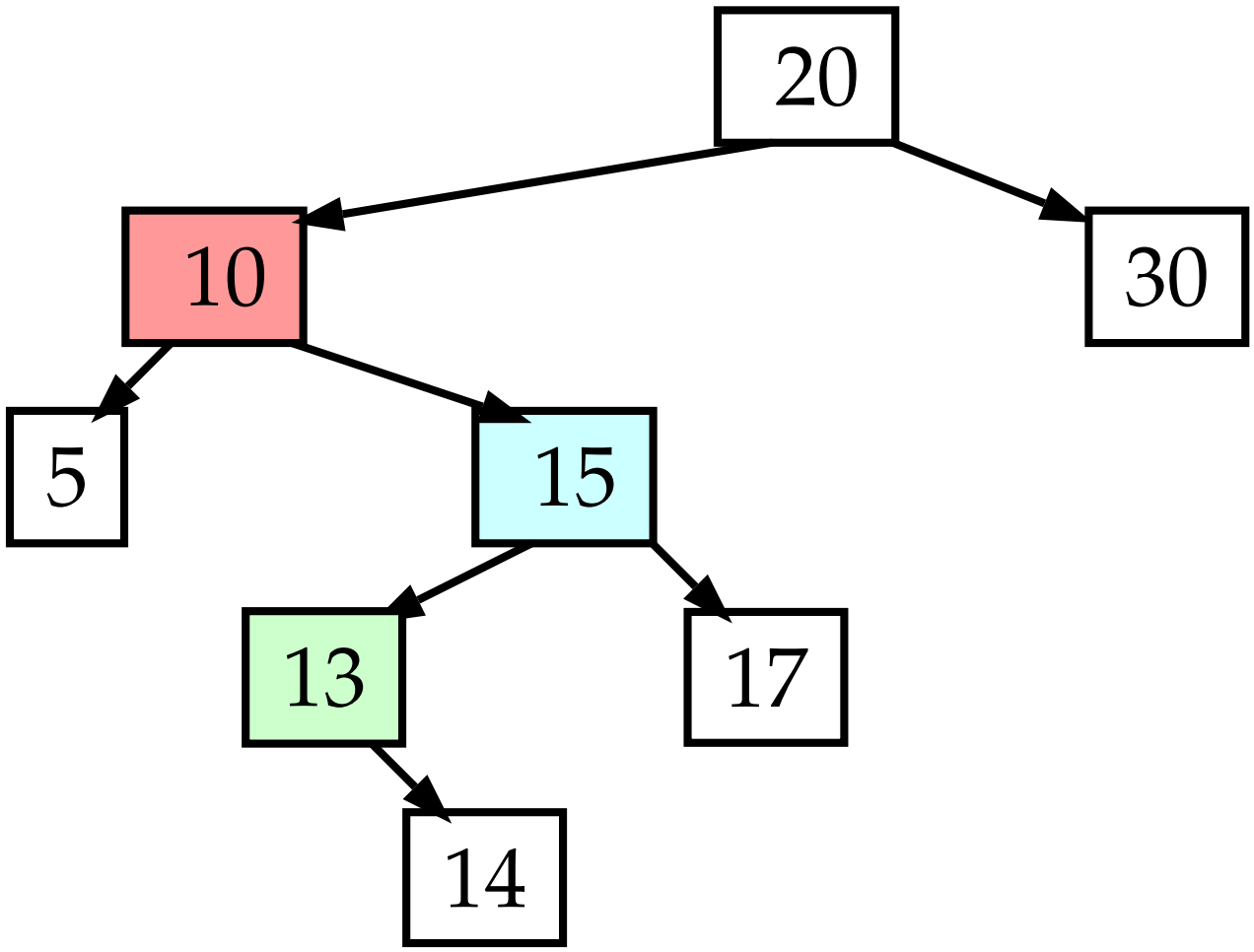
Sletting - ett barn



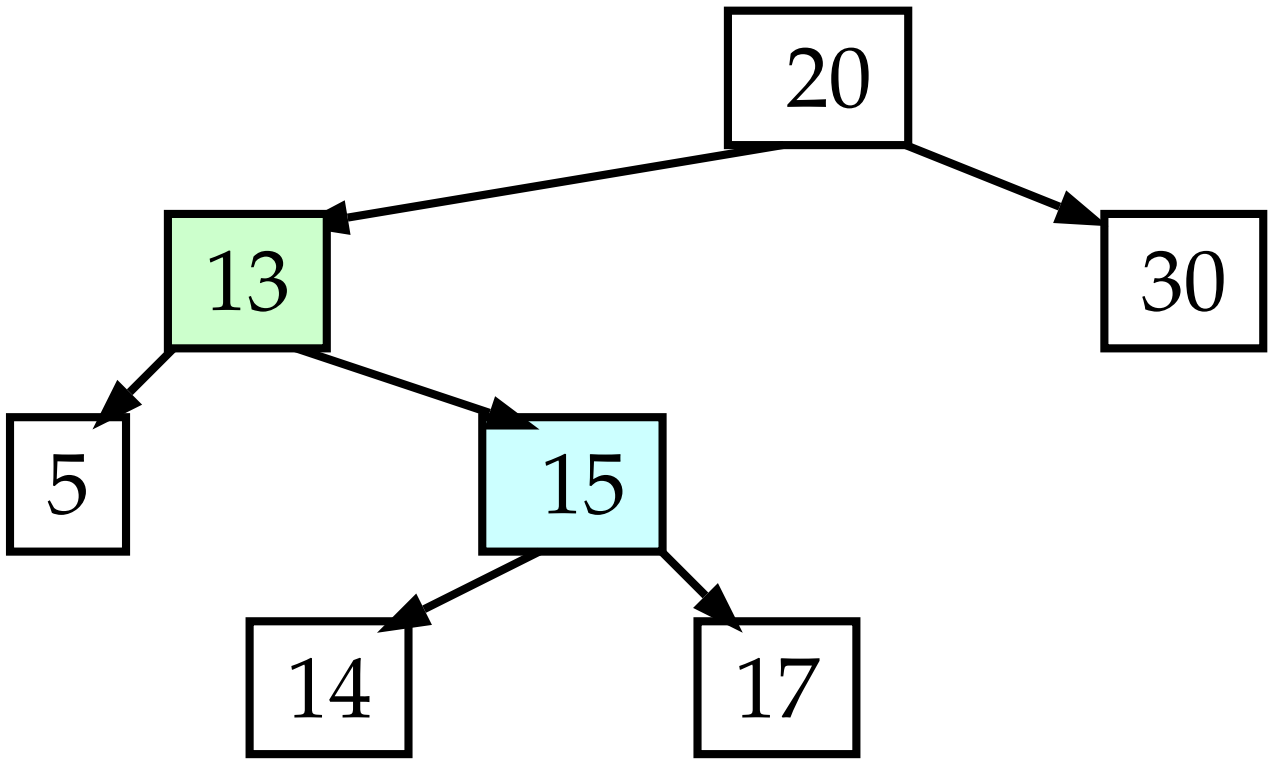
Sletting - ett barn



Sletting - to barn



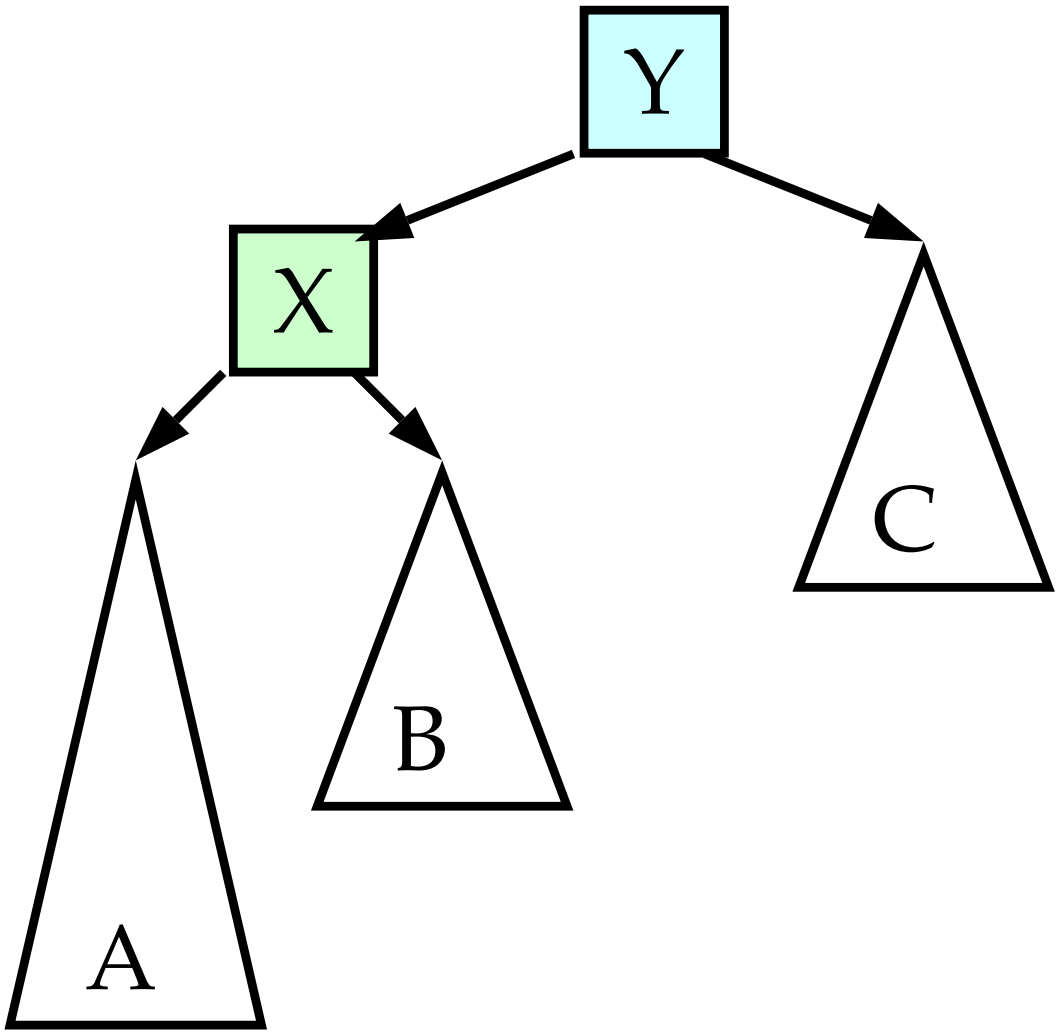
Sletting - to barn



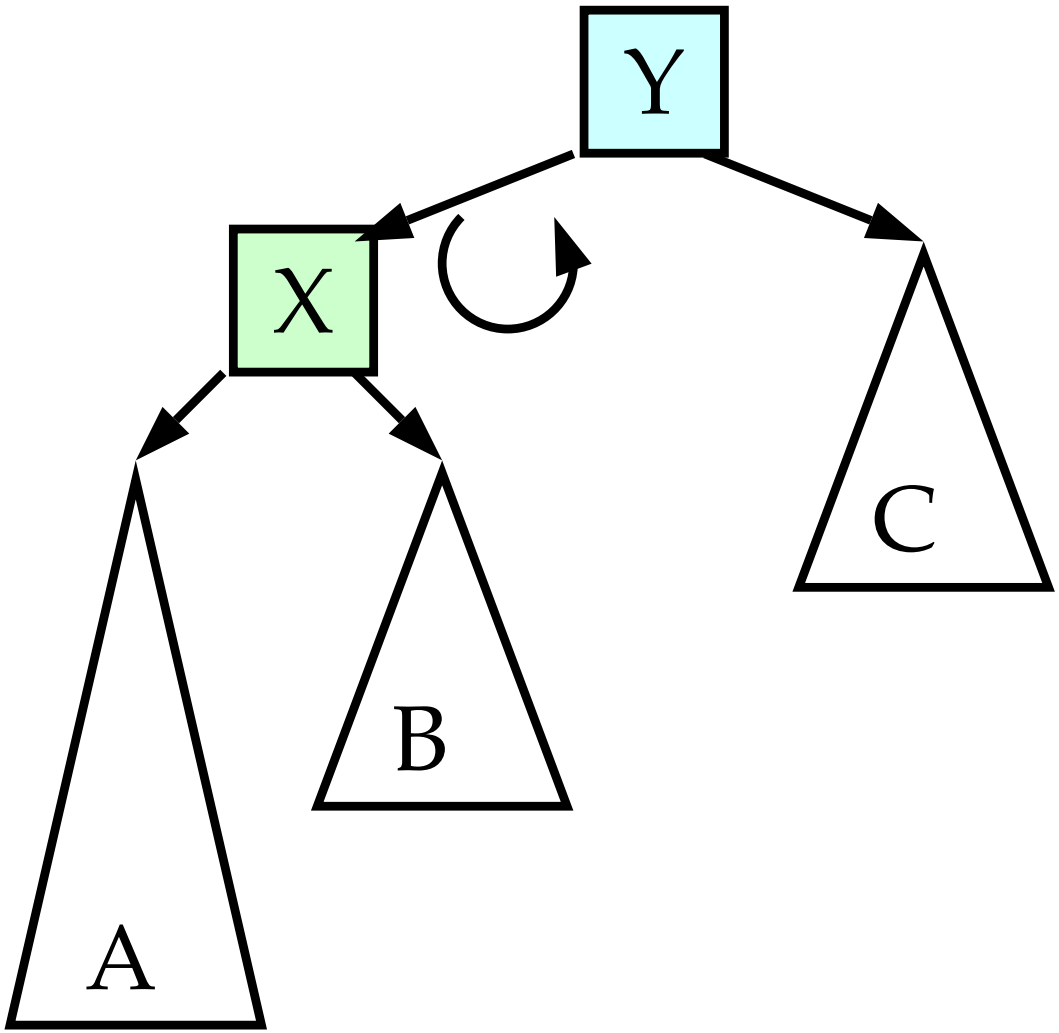
Sletting

```
1  Node delete(E e, Node branch)
2    if branch doesn't contain e
3      nextBranch = delete(e, nextBranch)
4      return branch
5
6    if left is null and right is null
7      return null
8
9    if branch has a single child
10     return child
11
12   if branch has two children
13     return innermost child of one of them
```

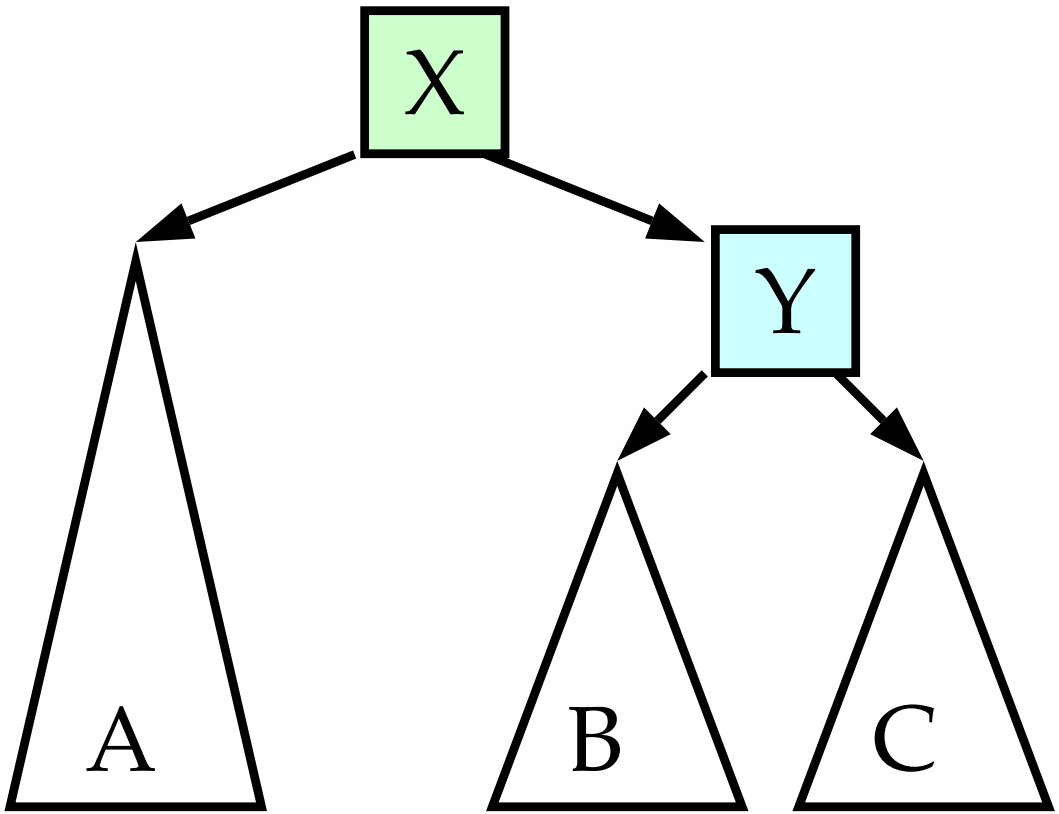
Enkeltrotasjon



Enkeltrotasjon



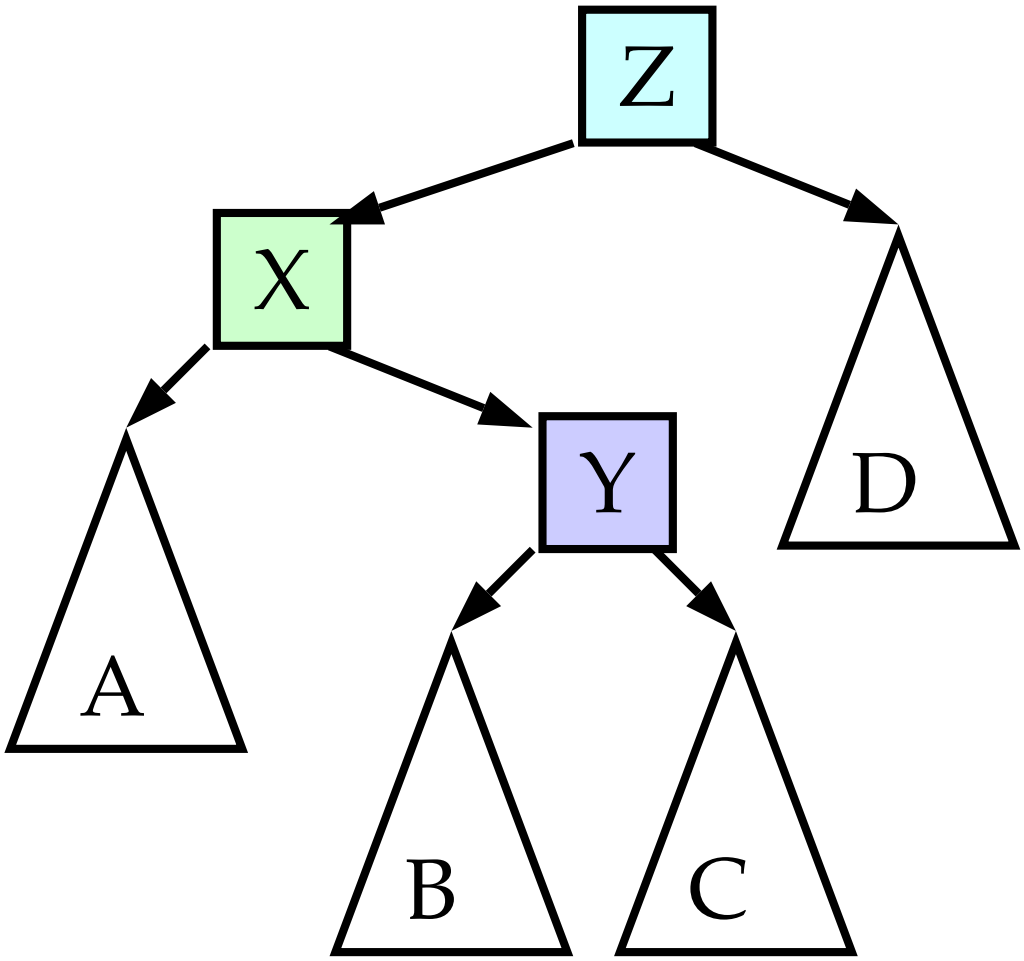
Enkeltrotasjon



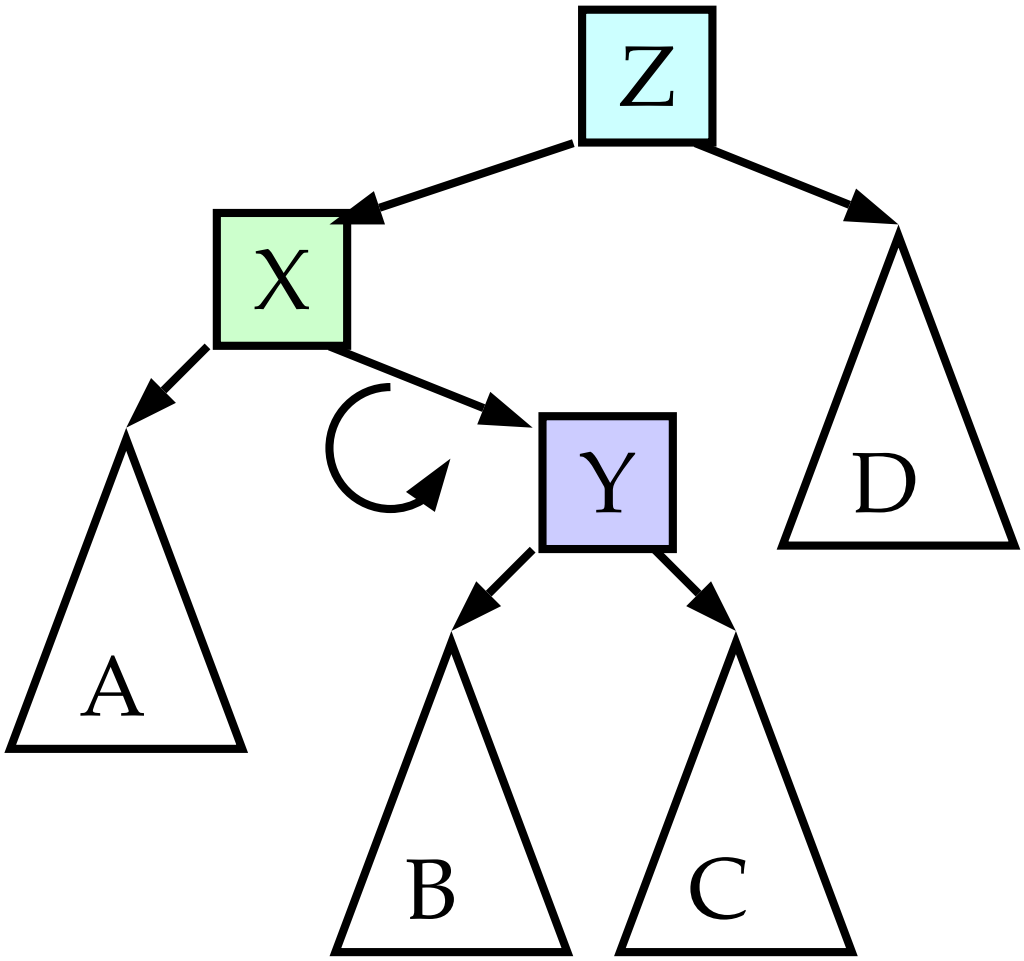
Enkeltrotasjon

```
1 Node rotateWithLeftChild(Node Y)
2    Node X = Y.left;
3    Y.left = X.right;
4    X.right = Y;
5    // Update X and Y's height variables
6    return X;
```

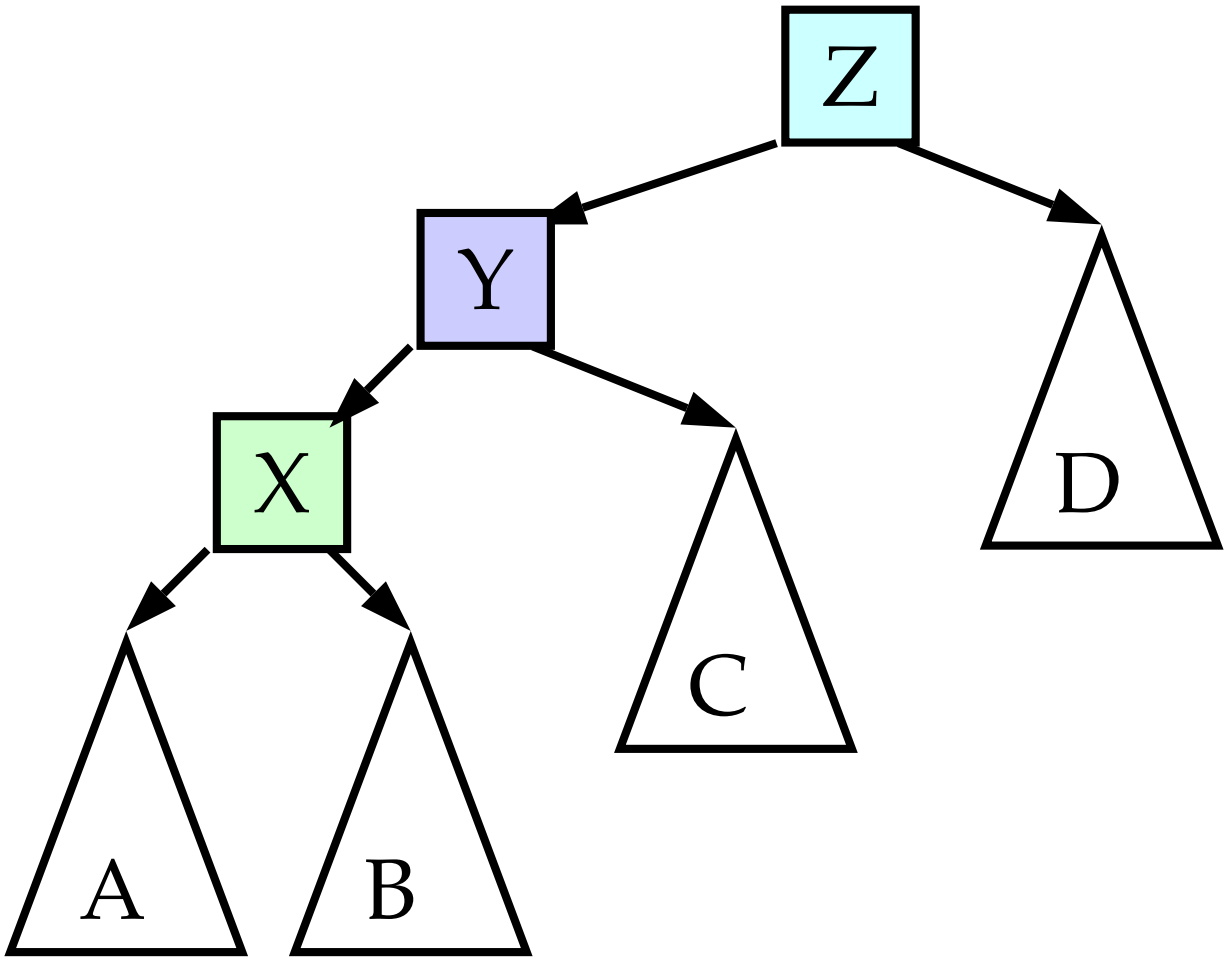
Dobbeltrotasjon



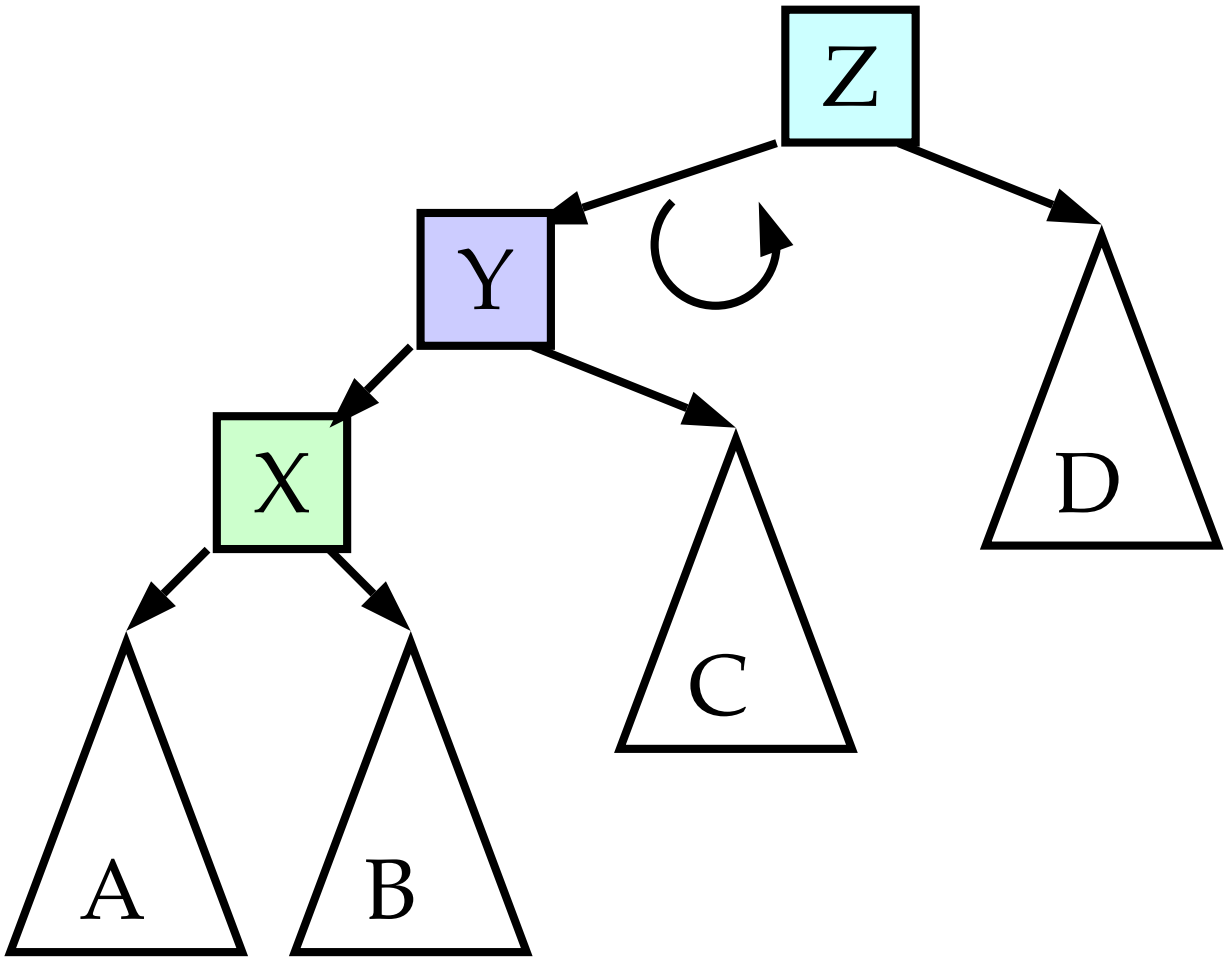
Dobbeltrotasjon



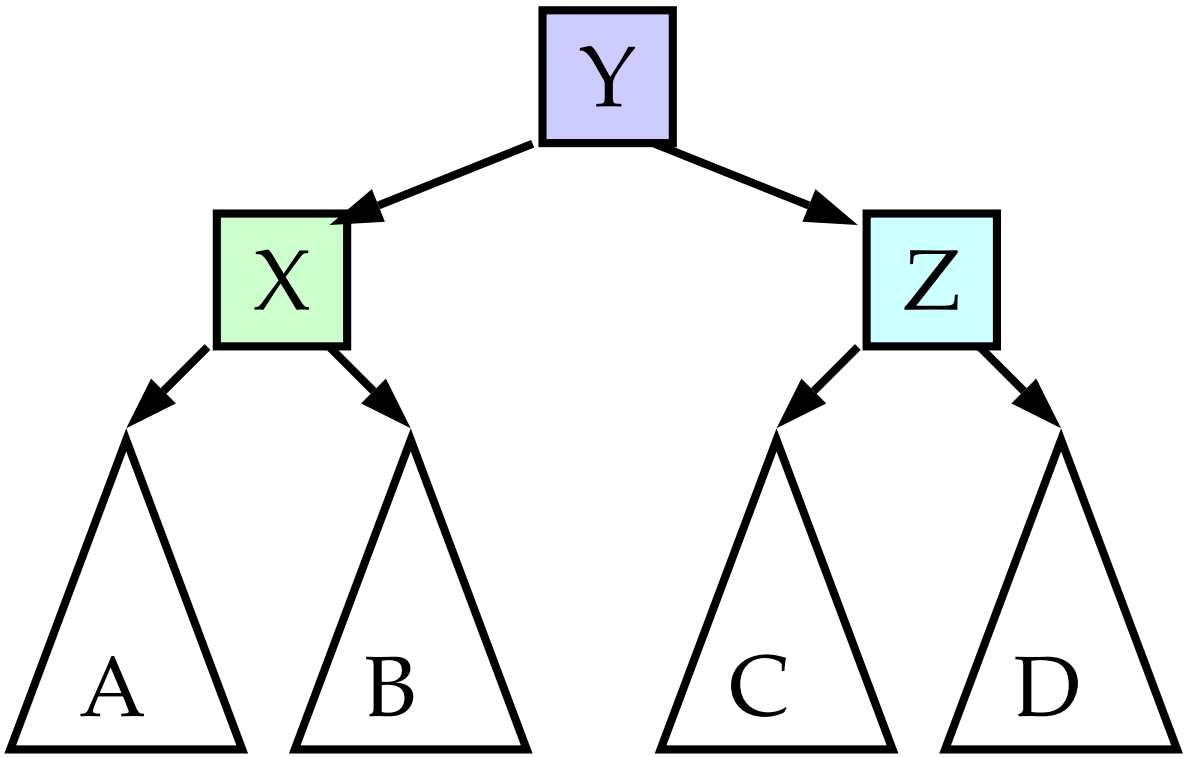
Dobbeltrotasjon



Dobbeltrotasjon



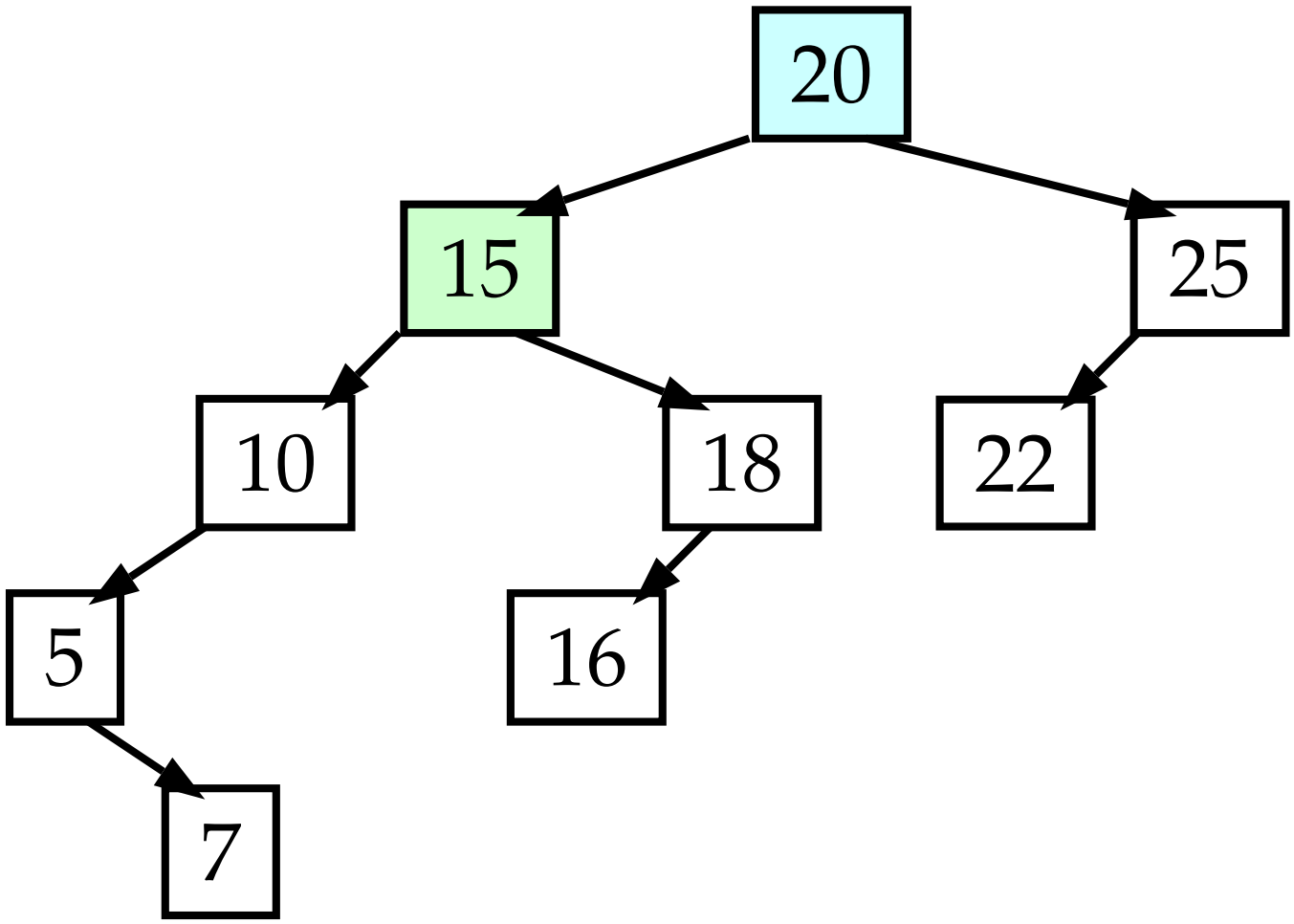
Dobbeltrotasjon



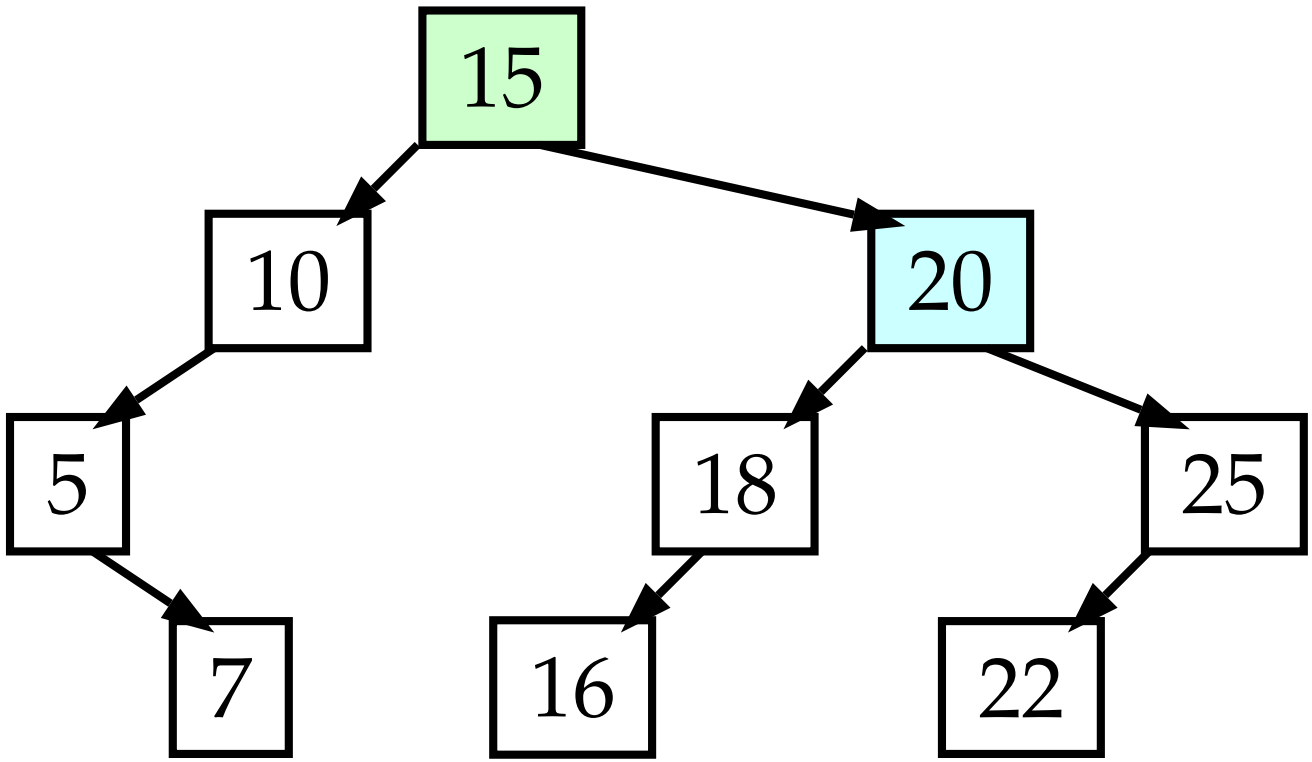
Dobbeltrotasjon

```
1 Node doubleRotateWithLeftChild(Node Z)
2   // Z.left is X
3   Z.left = rotateWithRightChild(Z.left)
4   return rotateWithLeftChild(Z)
```

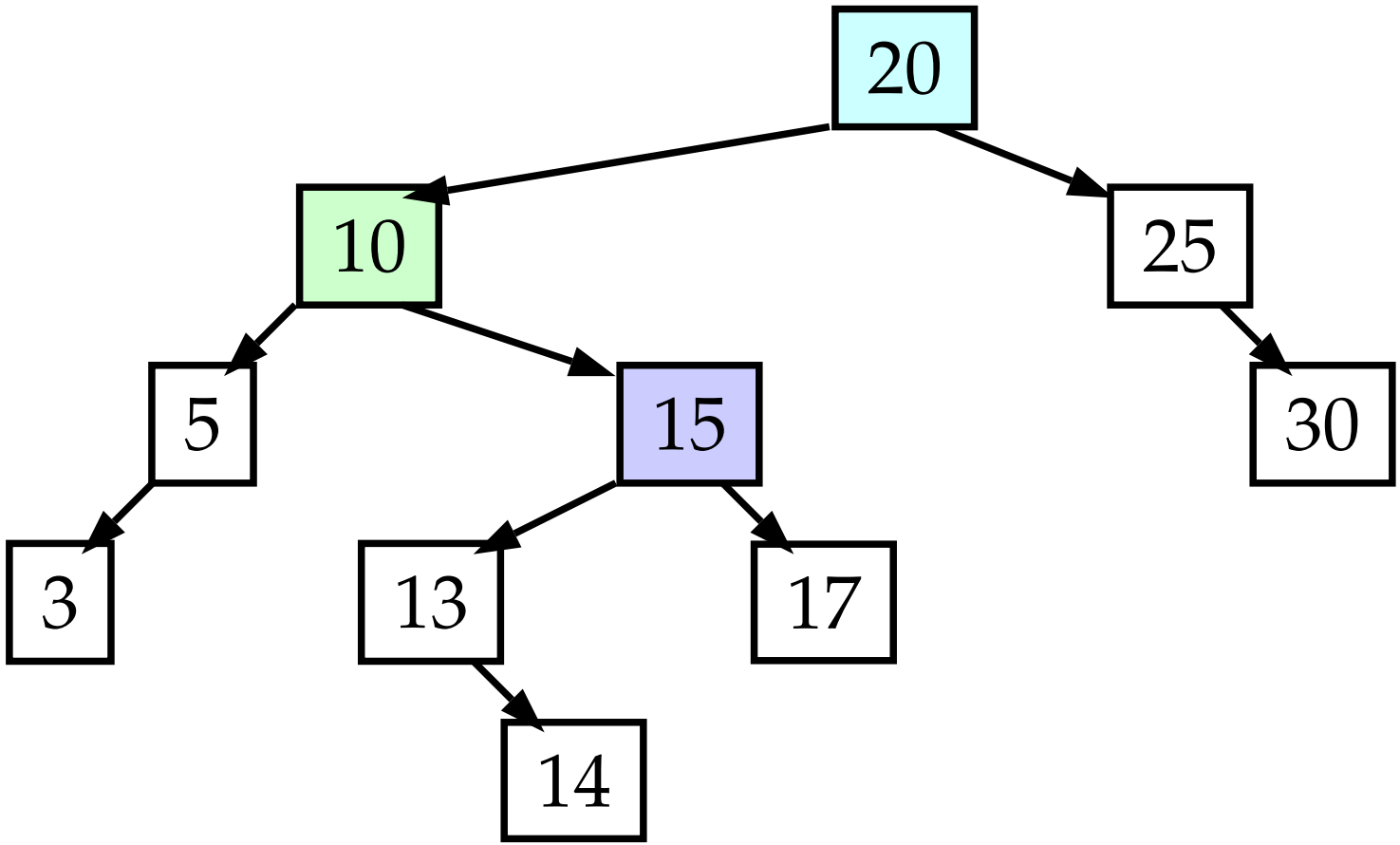
Enkeltrotasjon - eksempel



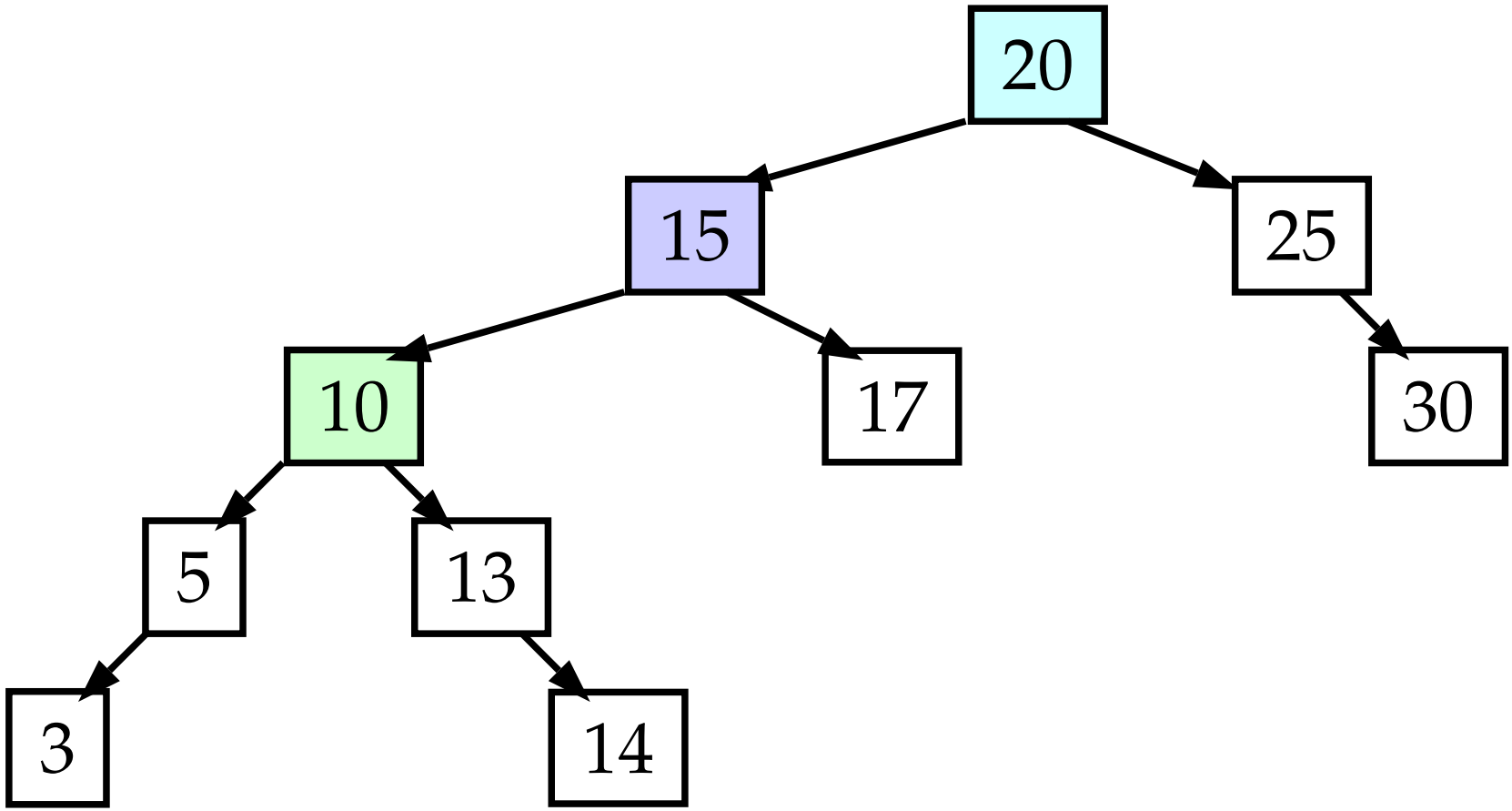
Enkeltrotasjon - eksempel



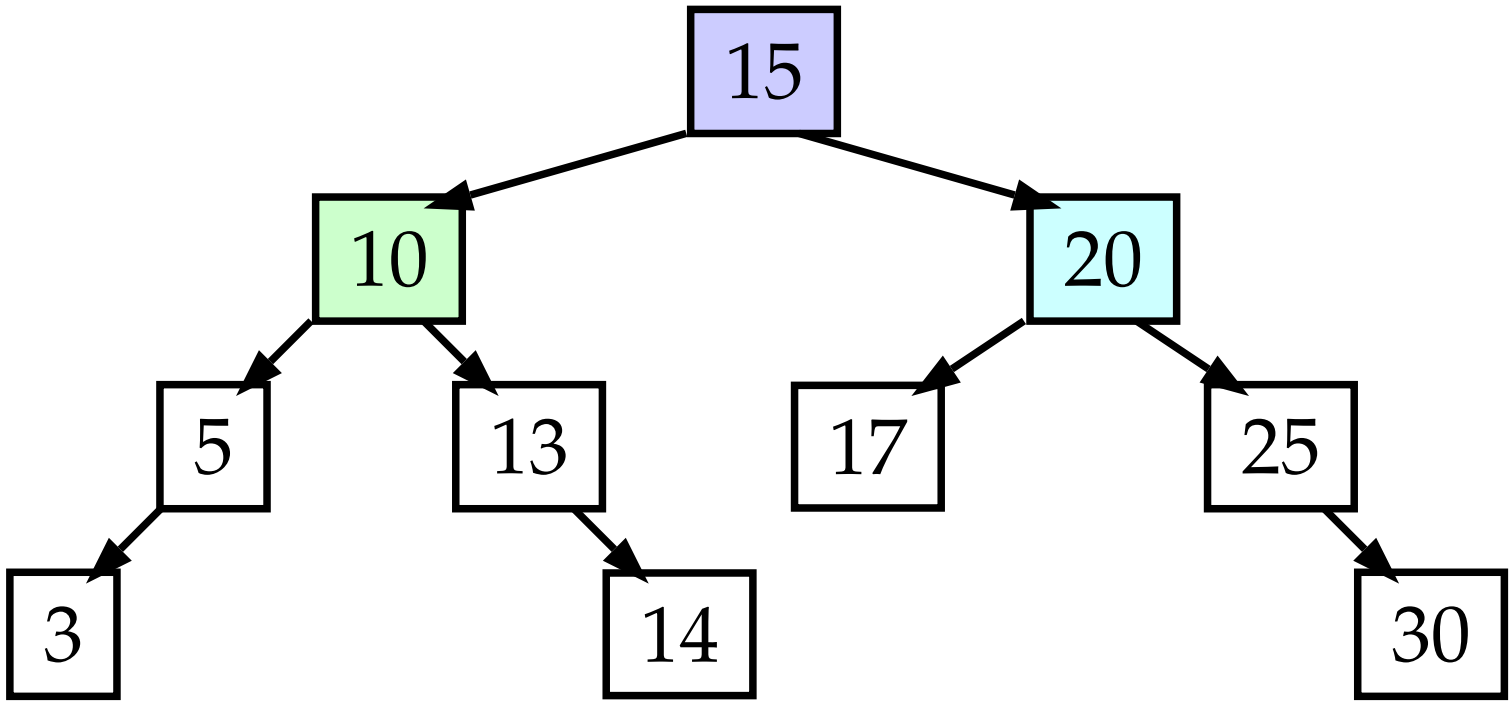
Dobbeltrotasjon - eksempel



Dobbeltrotasjon - eksempel



Dobbeltrotasjon - eksempel



Innsetting

```
1  Node insert(E in, Node branch)
2    if branch is null
3      return new Node(in)
4
5    if in is less than branch
6      branch.left = insert(in, branch.left)
7      if left branch is 2 higher than right branch
8        if in was placed in left branch's left branch
9          branch = rotateWithLeftChild(branch)
10       else
11         branch = doubleRotateWithLeftChild(branch)
12   else if n is greater than branch
13     // Do the same as above, but mirrored
14     // Do nothing for duplicates
15
16   branch.height = Math.max(left branch's height,
17                            right branch's height)
18   return branch // Branch might have changed
```